2.2 A Simple C Program: Printing a Line of Text

```c
/* Fig. 2.1: fig02_01.c */
#include <stdio.h>
/* function main begins program execution */
int main()
{
  printf( "Welcome to C!\n" );
  /* indicate that program ended successfully */
  return 0;
 /* end function main */
}
```

Comments

- Text surrounded by /* and */ is ignored by computer
- Used to describe program

• `#include <stdio.h>`
  - Preprocessor directive
    • Tells computer to load contents of a certain file
  - `<stdio.h>` allows standard input/output operations

---

2.2 A Simple C Program: Printing a Line of Text

• `int main()`
  - C programs contain one or more functions, exactly one of which must be `main`
  - Parenthesis used to indicate a function
  - `int` means that `main"returns" an integer value
  - Braces ({ and }) indicate a block
    • The bodies of all functions must be contained in braces
2.2 A Simple C Program: Printing a Line of Text

- `printf( "Welcome to C!\n" );`
  - Instructs computer to perform an action
    - Specifically, prints the string of characters within quotes (" ")
  - Entire line called a statement
    - All statements must end with a semicolon (;)
  - Escape character (\)
    - Indicates that printf should do something out of the ordinary
    - \n is the newline character

---

<table>
<thead>
<tr>
<th>Escape Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>\n</code></td>
<td>Newline: Position the cursor at the beginning of the next line.</td>
</tr>
<tr>
<td><code>\t</code></td>
<td>Horizontal tab: Move the cursor to the next tab stop.</td>
</tr>
<tr>
<td><code>\a</code></td>
<td>Alert: Sound the system bell.</td>
</tr>
<tr>
<td><code>\b</code></td>
<td>Backslash: Insert a backslash character in a string.</td>
</tr>
<tr>
<td><code>\&quot;</code></td>
<td>Double quote: Insert a double quote character in a string.</td>
</tr>
</tbody>
</table>

Fig. 2.2 Some common escape sequences.
2.2 A Simple C Program: Printing a Line of Text

- `return 0;`
  - A way to exit a function
  - `return 0`, in this case, means that the program terminated normally
- Right brace `{`
  - Indicates end of `main` has been reached

- The only action done by this programme is to write
  - Welcome to C!
- to the output device screen
/* Fig. 2.3: fig02_03.c */

Printing on one line with two printf statements */

#include <stdio.h>

/* function main begins program execution */

int main()
{
    printf("Welcome to C!\n");
    printf("\n");
    return 0; /* indicate that program ended successfully */
} /* end function main */

Program Output

Welcome to C!

/* Fig. 2.4: fig02_04.c */

Printing multiple lines with a single printf */

#include <stdio.h>

/* function main begins program execution */

int main()
{
    printf("Welcome to C!\n");
    return 0; /* indicate that program ended successfully */
} /* end function main */

Welcome to C!
/* Fig. 2.5: fig02_05.c

Addition program */

#include <stdio.h>

/* function main begins program execution */

int main()
{
    int integer1; /* First number to be input by user */
    int integer2; /* Second number to be input by user */
    int sum; /* Variable in which sum will be stored */

    printf("Enter first integer\n"); /* Prompt */
    scanf("%d", &integer1); /* Read an integer */

    printf("Enter second integer\n"); /* Prompt */
    scanf("%d", &integer2); /* Read an integer */

    sum = integer1 + integer2; /* Assign total to sum */

    printf("Sum is %d\n", sum); /* Print sum */

    return 0; /* Indicate that program ended successfully */
}